09/663568

In the Claims:

1. (Currently Amended) A method for forming a skeleton tool usable for generating a test suite for a verification system for verifying a Procedure Interface Under Test (PIUT), the method comprising the steps of:

decomposing existing test suites, the test suites having automatically generated components and manually developed components and being written in its test suite implementation language;

defining one or more standard schemes of procedure testing based on the decomposition of the test suites;

providing a skeleton description for each scheme in a skeleton definition language adapted to function as a directive for control of a transformation; and

transforming the skeleton description for each scheme into a skeleton tool for generating the test suite of the scheme.

2. (Original) The method as claimed in claim 1, wherein

the decomposing step includes a step of identifying an invariant test suite part written in the test suite implementation language; and

the skeleton description providing step includes a step of creating an invariant test suite description based on the identified invariant test suite part.

3. (Original) The method as claimed in claim 1, wherein

the decomposing step includes a step of identifying a skeleton parameter; and

the skeleton description providing step includes a step of creating a skeleton parameter identifier based on the identified skeleton parameter.

4. (Original) The method as claimed in claim 3, wherein the skeleton parameter includes a text string or an integer.

5. (Original) The method as claimed in claim 3, wherein the skeleton parameter includes an array of text strings or integers.

09/663568

6. (Original) The method as claimed in claim 1, wherein

the decomposing step includes a step of identifying a repetitive part of the test suite; and

the skeleton description providing step includes a step of creating a repetitor describing the identified repetitive part of the test suite.

7. (Original) The method as claimed in claim 6, wherein the repetitor is written in a macro-language.

8. (Currently Amended) The method as claimed in claim 1, wherein

the decomposing step includes a step of identifying a variant of the test suite; and

the skeleton description providing step ~~include~~ includes a step of creating a variant descriptor describing the identified variants of the test suite.

9. (Original) The method as claimed in claim 8, wherein the variant descriptor is written in a macro-language.

10. (Original) The method as claimed in claim 1, wherein

the decomposing step includes a step of identifying a manually-developed component of the test suite; and

the skeleton description providing step includes a step of creating a slot descriptor describing a slot for receiving a component corresponding to the identified manually-developed component of the test suite.

11. (Original) The method as claimed in claim 10, wherein the slot descriptor is written in a macro-language.

12. (Original) The method as claimed in claim 10, wherein the slot descriptor provides rigorously defined semantics for manual slot filling.

13. (Original) The method as claimed in claim 1, wherein

09/663568

the decomposing step includes a step of identifying an automatically-generated component of the test suite; and

the skeleton description providing step includes a step of creating a slot descriptor describing a slot for receiving a component corresponding to the identified automatically-generated component of the test suite.

14. (Original) The method as claimed in claim 13, wherein the slot descriptor is written in a macro-language.

15. (Original) The method as claimed in claim 1, wherein the defining step includes a step of separating procedure testing into different types, each type corresponding to each standard scheme, based on dependency of parameters.

16. (Original) The method as claimed in claim 1, wherein the defining step includes a step of separating procedure testing into different types, each type corresponding to each standard scheme, based on sequences of testing of a group of procedures.

17. (Original) The method as claimed in claim 1, wherein the transforming step includes a step of creating a file containing values of skeleton parameters and components for slot filing.

18. (Currently Amended) A system for forming a skeleton tool usable for generating a test suite for a verification system for verifying a Procedure Interface ~~under~~ Under Test (PIUT), the system comprising:

a decomposer for decomposing test suites, the test suites having automatically generated components and manually developed components and being written in its test suite implementation language, the decomposed test suites being used to define one or more standard schemes of procedure testing;

a skeleton describer for providing a skeleton description for each scheme in a skeleton definition language adapted to function as a directive for control of a transformation; and

a skeleton transformer for transforming the skeleton description for each scheme into a skeleton tool for generating the test suite of the scheme.

09/663568

19. (Original) The system as claimed in claim 18, wherein the decomposer includes a feature identifier for identifying particular features of the test suites.

20. (Original) The system as claimed in claim 18, wherein the skeleton describer includes a creator for creating parts of skeleton description based on the identified features of the test suites.

21. (Original) The system as claimed in claim 19, wherein the feature identifier includes:

an invariant test suite part identifier for identifying an invariant test suite part written in the test suite implementation language;

a skeleton parameter identifier for identifying a skeleton parameter;

a repetitive part identifier for identifying a repetitive part of the test suite;

a variant identifier for identifying a variant of the test suite;

a manually-developed component identifier for identifying a manually-developed component of the test suite; and

an automatically-developed component identifier for identifying an automatically-generated component of the test suite.

22. (Original) The system as claimed in claim 21, wherein the skeleton describer includes:

an invariant description creator for creating an invariant test suite description based on the identified invariant test suite part;

a parameter identifier creator for creating a skeleton parameter identifier based in the identified skeleton parameter;

a repetitor creator for creating a repetitor describing the identified repetitive part of the test suite;

a variant descriptor creator for creating a variant descriptor describing the identified variants of the test suite;

a first slot descriptor creator for creating a slot descriptor describing a slot for receiving a component corresponding to the identified manually-developed component of the test suite; and

a second slot descriptor creator for creating a slot descriptor describing a slot for receiving a component corresponding to the identified automatically-generated component of the test suite.

09/663568

23. (Original) The system as claimed in claim 18, wherein the scheme defining means has a function for separating procedure testing into different types, each type corresponding to each standard scheme, based on dependency of parameters.

24. (Currently Amended) A method for generating test suite sources for a verification system for verifying a Procedure Interface Under Test (PIUT), the method comprising steps of:

decomposing existing test suites, the test suites having automatically generated components and manually developed components and being written in its test suit implementation language;

defining one or more standard schemes of procedure testing based on the decomposition of the test suites;

providing a skeleton description for each scheme in a skeleton definition language adapted to function as a directive for control of a transformation;

transforming the skeleton description for each scheme into a skeleton tool for generating the test suite of the scheme;

generating a test suite template by executing the skeleton tool using a specification of the PIUT; and

filling the test suite template using manually written parts of the test suite to ~~generating~~ generate test suite sources.

25. (Original) The method as claimed in claim 24 further comprising a step of compiling the test suite sources into a test suite in implementation language.